Black-box Optimization of Sensor Placement with Elevation Maps and Probabilistic Sensing Models

Vahab Akbarzadeh, Christian Gagné, and Marc Parizeau
Laboratoire de vision et systèmes numériques
Département de génie électrique et de génie informatique
Université Laval, Québec (Quebec), Canada G1V 0A6
Emails: vahab.akbarzadeh.1@ulaval.ca,
{christian.gagne,marc.parizeau}@gel.ulaval.ca

Mir Abolfazl Mostafavi
Centre de recherche en géomatique
Département des sciences géomatiques
Université Laval, Québec (Quebec), Canada G1V 0A6
Email: mir-abolfazl.mostafavi@scg.ulaval.ca

***Abstract*—This paper proposes a framework for the optimization of sensor placement. Traditional schemes rely on simple sensor behaviours and environmental factors. The consequences of these oversimplifications are unrealistic simulation of sensor performance and, thus, suboptimal sensor placement. In this paper, we develop a novel framework to tackle the sensor placement problem using a probabilistic coverage and corresponding membership functions for sensing range and sensing angle, which takes into consideration sensing capacity probability as well as critical environmental factors such as terrain topography. We then implement several optimization schemes for sensor placement optimization, including simulated annealing, L-BFGS, and CMA-ES.**

I. Introduction

Wireless Sensor Networks (WSN) are built from a collection of small, inexpensive sensor devices, where each sensor has limited sensing, storage, processing, and communication capabilities. With the recent proliferation of Micro-Electro-Mechanical Systems (MEMS), we have seen a rapid increase of interest in WSNs [1], where sensors can make measurements in the environment and gather information for end-users.

There are a number of fundamental issues that should be addressed for effective exploitation of WSNs, such as localization, tracking, security, data aggregation, and placement. Placement is an example of a more general problem of configuring sensor parameters. Depending on the application of WSN and the sensor type being used, each sensor has a number of variable parameters that must be determined, e.g., latitude and longitude, orientation, and operating range of each sensor in the placement problem. There are four main issues which should be taken into consideration for an optimal placement of WSNs, namely, performance maximization, reliability maximization, energy saving, and cost minimization.

Considering a region of interest monitored by sensors, overall performance of the network is measured by coverage [2]–[7]. In general, one of the basic requirements for a WSN is that each location in a region of interest should be within the sensing range of at least one of the sensors. An alternative approach is to have a region of interest covered simultaneously by at least K sensors [2], [4].

Although many deterministic methods have been explored to address the problem of coverage, traditional sensor placement strategies often rely on oversimplified sensor models and environmental factors [2], [4], [5], [8], [9]. These deterministic approaches cannot deal with environmental factors such as terrain topography, and usually assume an omnidirectional disk sensing model for each sensor. In fact, under the assumptions of uniform disk sensing model, it has been shown that optimal coverage can be deterministically achieved with a regular placement of sensors [3], [7], [10]. Similar results have also been reported when multiple coverage of the target area is required [2], [4], [5], [10].

The direct consequence of such oversimplifications is that the theoretical perfect coverage shown in deterministic methods may not hold true in practice. This may be the result of a number of causes. First, most sensor placement optimization methods assume that sensors are placed on a 2D plane, without taking into account the topography of the terrain [3], [7], [10]. Second, many methods assume that sensors have omnidirectional sensing capabilities [11]. But antennas and microphones have non-uniform 3D reception fields that depend on factors like orientation, distance, and other environmental factors [11], cameras have narrow field of views, etc. Third, sensors usually do not have a binary coverage range as it is often assumed in traditional sensor placement methods [3], [10]. Although some probabilistic sensing range models [3], [7], [10], [12], [13] and sensing models with irregular sensing ranges [14] have been proposed, they all operate on a 2D flat space and are omnidirectional. Recently, Topcuoglu et al. [15] proposed a new formulation for deployment of the sensors in a synthetically generated 3D environment. Although the proposed approach makes several realistic assumptions regarding the modelling of the environment and sensors, it assumes a binary sensing area for each sensor inside the environment. The combinational effects of terrain variations, blind points sensing angle or irregular sensing range, and probabilistic sensing property of sensors have never been studied before.

The limitations of the deterministic placement methods are thus obvious, and the 100% coverage that they claim is often over-estimated. This issue is critical because it further complicates the problem of sensor placement: while a WSN may seem to satisfy the requirements to achieve full coverage on a target area using a deterministic method, the deployers of such a network have no means of ensuring that this coverage is truly effective in a real environment.

Facing this challenge, we follow a more flexible non-deterministic avenue. Our aim is to optimize sensor placement using topographic information of the terrain and probabilistic sensor modelling. Our approach differs from previous methods in the following three ways: 1) deterministic schemes only consider 2D environments and ignore the effects of elevation, whereas our method takes into account the 3D terrain information; 2) deterministic schemes usually assume omnidirectional sensors, whereas our method allows for constraints to be applied on sensors, such as limited sensing angles and range; 3) deterministic schemes implement mainly binary coverage, i.e., a point can only be classified as covered or uncovered, whereas our method applies probabilistic coverage in both sensing distance and sensing angle.

In order to tackle these problems, we develop a novel framework for sensor placement that takes into account the above mentioned issues, and then compare this approach with some classical optimization algorithms. This paper extends our previous work [16] by proposing directional and probabilistic sensor models, and testing the optimization with other methods, that is simulated annealing and L-BFGS.

The remainder of the paper is organized as follows. The proposed framework is presented in the next section (Sec. II), followed by a presentation of the optimization methods in Sec. III, including experimental protocol and results on sensor placement. We conclude the paper with a summary of results and perspectives (Sec. IV).

II. Proposed Framework for Sensor Placement

A. Coverage Definition in a Sensor Network

The sensing model depends on distance, orientation, and visibility. We first assume that all sensors are positioned at a certain constant height above the ground level. The sensor position is thus described by a 3D point $\mathbf{p} = (x, y, z)$, where (x, y) are free parameters, and $z = g(x, y)$ is constrained by the terrain elevation at position (x, y), as defined by a Geographic Information System (GIS). We further assume that the anisotropic properties of sensors are fully defined by a pan angle θ around the vertical axis (tilt angle is currently assumed to be zero). Given the GIS, a sensor network $N = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ of n sensors is thus fully specified by $3n$ free parameters $\mathbf{s}_i = (\mathbf{p}_i, \theta_i)$, $i = 1, 2, \ldots, n$, with $\mathbf{p}_i = (x_i, y_i)$.

Now the coverage $C(\mathbf{s}_i, \mathbf{q})$ of sensor $\mathbf{s}_i$ at point $\mathbf{q}$ in the environment can be defined as a function of distance $d(\mathbf{s}_i, \mathbf{q}) = \|\mathbf{p}_i - \mathbf{q}\|$, angle of view $a(\mathbf{s}_i, \mathbf{q}) = \theta_i - \angle(\mathbf{q} - \mathbf{p}_i)$, and visibility $v(\mathbf{s}_i, \mathbf{q})$ from the sensor:

$$C(\mathbf{s}_i, \mathbf{q}) = f[\mu_d(\|\mathbf{p}_i - \mathbf{q}\|), \mu_a(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)), v(\mathbf{p}, \mathbf{q}, g_{\text{elev}}(\mathbf{p}_i))], \tag{1}$$

where $\angle(\mathbf{q} - \mathbf{p}_i) = \arctan\left(\frac{y_{\mathbf{q}} - y_{\mathbf{p}_i}}{x_{\mathbf{q}} - x_{\mathbf{p}_i}}\right)$ is the pan angle of point $\mathbf{q}$ relative to $\mathbf{p}_i$. For $\mathbf{q}$ to be covered by sensor $\mathbf{s}_i$, we need to take into account its sensing range, that it is within its field of view, and its visibility, that is not blocked by any terrain obstacle such as hills. Let $\mu_d, \mu_a \in [0, 1]$ represent some membership functions of the first two conditions, then Equation 1 can be rewritten as a multiplication of these memberships:

$$C(\mathbf{s}_i, \mathbf{q}) = \mu_d(\|\mathbf{p}_i - \mathbf{q}\|) \cdot \mu_a(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)) \cdot v(\mathbf{p}_i, \mathbf{q}, g_{\text{elev}}(\mathbf{p}_i)). \tag{2}$$

Function $v(\mathbf{p}_i, \mathbf{q}, g_{\text{elev}}(\mathbf{p}_i))$ is usually binary. Given a sensor position $\mathbf{p}_i$ and sensor elevation $g_{\text{elev}}(\mathbf{p}_i)$, if the line of sight between sensor $\mathbf{s}_i$ and $\mathbf{q}$ is obscured, then we assume that the coverage cannot be met ($v = 0$), otherwise the visibility condition is fully respected ($v = 1$). Memberships μ_d and μ_a need to be defined according to their parameters. In our experiment, the elevation is set to $g_{\text{elev}}(\mathbf{p}_i) = g(x_i, y_i) + 1$, that is 1 meter above the ground at the position of the sensor.

At each position $\mathbf{q} \in \Xi$ of environment Ξ, the coverage for a single sensor is thus the multiplication of the three above conditions. Value $C = 1$ means full coverage, and $C = 0$ indicates no coverage. If more than one sensor covers $\mathbf{q}$, then the local network coverage C_l can be computed as follows:

$$C_l(N, \mathbf{q}) = 1 - \prod_{i=1,\ldots,n} (1 - C(\mathbf{s}_i, \mathbf{q})), \tag{3}$$

and the global coverage C_g can be:

$$C_g(N, \Xi) = \frac{1}{|\Xi|} \sum_{\mathbf{q} \in \Xi} C_l(N, \mathbf{q}). \tag{4}$$

Given an environment Ξ, the problem statement is thus to determine the sensor network placement N that maximizes $C_g(N, \Xi)$.

B. Probabilistic Membership Functions

The membership functions μ_d and μ_a can be defined as crisp functions, with value of 1 when the position is within a fixed sensing range or angle of view, and otherwise 0.

$$\mu_d(\|\mathbf{p}_i - \mathbf{q}\|) = \begin{cases} 1 & \|\mathbf{p}_i - \mathbf{q}\| \leq d_{max} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\mu_a(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)) = \begin{cases} 1 & (\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)) \in [-a, a] \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

However, such functions used in a coverage function provide essentially a binary 0/1 signal, which is not a good feedback to use for optimizing functions given its lack of information. Therefore we propose to use real-valued membership functions that provide a monotonically decreasing membership value over distance and relative angle between the position and sensor.

We propose the following membership function for distance:

$$\mu_d(\|\mathbf{p}_i - \mathbf{q}\|) = \frac{1}{1 + \exp\left[-\left(\frac{\alpha}{\|\mathbf{p}_i - \mathbf{q}\|} - \beta\right)\right]}, \tag{7}$$

with $\alpha \geq 0$ and β as parameters configuring the membership function. These parameters can be approximated using experimental observations on sensor behaviours. Figure 1 presents the μ_d function for different values of α and β.

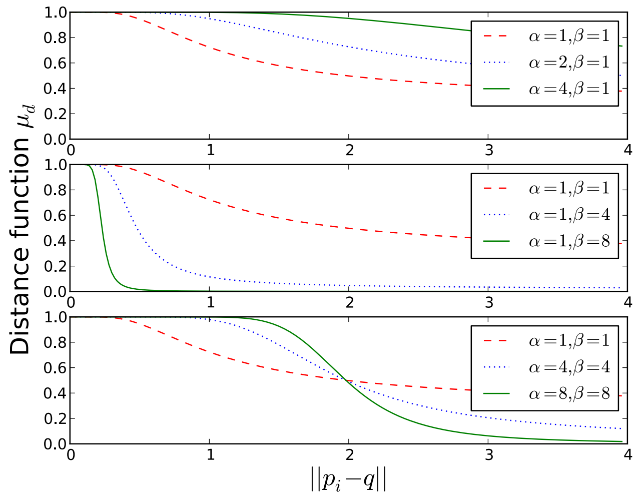


Fig. 1. The effects of variations of α and β on $\mu_d(||\mathbf{p}_i - \mathbf{q}||)$.

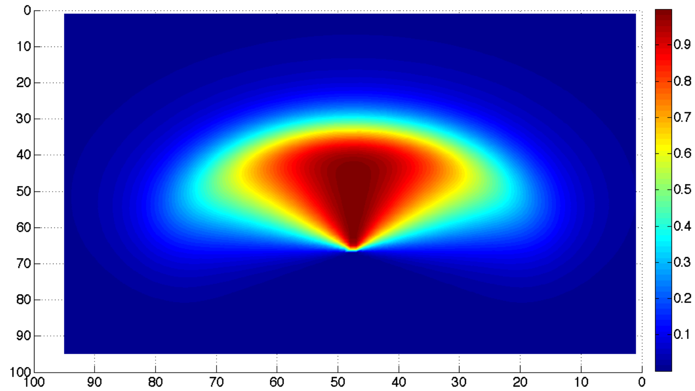


Fig. 3. Probabilistic coverage model of a sensor. Assuming that a sensor is positioned at (48,66) heading upward, the colour shows different degrees of coverage for points inside the map.

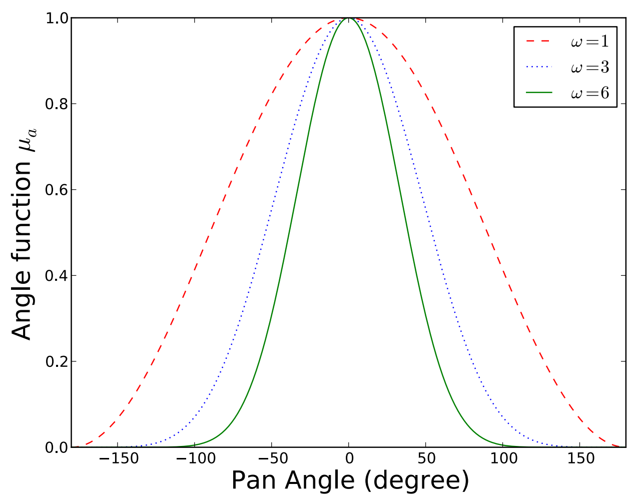


Fig. 2. The effects of variations of ω on $\mu_a(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i))$.

As for the membership function for an angle, we propose the following function:

$$\mu_a(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)) = \left(\frac{cos(\theta_i - \angle(\mathbf{q} - \mathbf{p}_i)) + 1}{2}\right)^{\omega}, \quad (8)$$

where $\omega \geq 1$. Figure 2 shows the change in membership function μ_a with three values of ω. Here, we assume that the pan angle for each sensor is adjustable and the visibility along the tilt angle is complete from -90^o to 90^o.

For a reasonable model of a sensor, we propose to use as parameters $\alpha = 350$, $\beta = 10$, and $\omega = 3$. With these values, the sensors have an effective sensing range of 30m and sensing angle of 120^o (see Figure 3 for an illustration of the coverage obtained).

III. Implemented Optimization Methods

From this framework for sensor placement optimization, we compare four sensor placement schemes: a deterministic approach found in the WSN literature, an adaptation of Simulated Annealing (SA) for sensor placement, the limited-memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS), and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), an evolutionary algorithm for real-valued optimization. The deterministic approach is purely geometrical and does not take into account the framework proposed in the previous section. As for the three other optimization methods, they have been applied on a real-valued vector composed of three values per sensor (x_i, y_i, θ_i), so as to maximize the global sensor network coverage $C_g(N, \Xi)$ over a given elevation map Ξ:

$$N = \{(x_1, y_1, \theta_1), (x_2, y_2, \theta_2), \ldots, (x_n, y_n, \theta_n)\}, \quad (9)$$
$$N^* = \operatorname*{argmax}_N C_g(N, \Xi). \quad (10)$$

We briefly explain each method in the following sections.

A. Deterministic Sensor Placement

The deterministic method has been shown to achieve full coverage on the Cartesian plane [7], [17]. Figure 4 illustrates this placement pattern, where sensors are organized in layers of horizontal stripes. Assuming sensors with sensing range r_s, they are simply distributed $\sqrt{3}r_s$ apart on every stripe, and the stripes are themselves separated from one another by $\frac{3}{2}r_s$. Furthermore, the stripes are interleaved to form a triangular lattice pattern. This approach does not take the terrain into consideration.

As presented in Figure 3, the pan angle coverage of each sensor is roughly 120^o. Therefore, to obtain an omnidirectional coverage at each position (because the deterministic approach assumes that all sensors have omnidirectional coverage), we place three sensors facing 120^o degrees apart from each other, at each position specified by the deterministic method.

B. Simulated Annealing Method on Single Sensor

SA [18] is a heuristic optimization algorithm. With a generic probabilistic heuristic approach, simulated annealing may escape local optimum and converge to global optimum, and thus

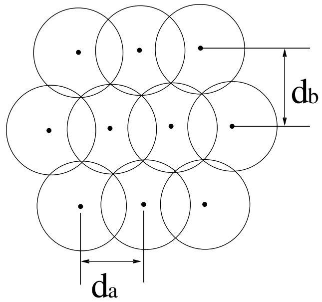


Fig. 4. Pattern of the deterministic method [7], [17] implemented in the paper, where $d_a = \sqrt{3}r_s$, $d_b = \frac{3}{2}r_s$, and r_s is sensing range for a sensor. Circles are sensor sensing ranges, and dots are sensor positions.

may be more effective for a global optimization problem of a given function in a large search space. Our implementation of SA is described in Figure 5. It requires the definition of the temperature function ($\mathrm{temperature}(t)$), and the setting of two parameters (M, σ)

- Parameter M defines the maximum iterations for simulated annealing. The larger the M, the more time consuming the optimization, and the more likely the global optimum can be reached.
- σ defines candidate generator, i.e., the size of neighbourhood where subsequent solutions will be generated. An essential requirement for σ is that it must provide a sufficiently short path from the initial state to any state which may be the global optimum. Another issue is that σ should be selected so that the search path avoids becoming trapped in a local minimum, i.e., it must be large enough to cross local minima in an effort to reach the global optimum.
- Temperature function ($\mathrm{temperature}(t)$) defines the probability of accepting a move in simulated annealing. Initially, the $\mathrm{temperature}(t)$ is set to a high value, then it is decreased at each step according to some annealing schedule, and finally ends with $\mathrm{temperature}(t) \to 0$ towards the end of the allotted maximum steps, M. When the temperature is high, the probability of accepting a move will be high. When the cooling rate is low, the probability of accepting a move decreases. The idea is that the system is expected to wander initially towards a broad region of the search space containing good solutions, ignoring small features of the energy function; then drift towards low-energy regions that become narrower and narrower. To satisfy the conditions above, the $\mathrm{temperature}(t)$ is defined as an exponential decay function as follows:

$$\mathrm{temperature}(t) = \frac{1}{2}\exp\left[-\frac{2\ln 2 \times t}{M}\right] \qquad (11)$$

C. Limited-memory BFGS method

BFGS [19] is a numerical optimization method for solving non-linear optimization problems. This method is an example of Quasi-Newton optimization methods, which find the stationary point of a function without computing the Hessian matrix of the objective function. It rather updates an estimate of the inverse Hessian matrix. Limited-memory BFGS (L-BFGS) which stores few past inverse Hessian matrix updates instead of the full matrix. Therefore, L-BFGS is a black-box deterministic optimization method well suited for problems with a high number of variables.

Initialize sensor network with random positions uniformly distributed in placement domain (assuming domain to be in $\mathbf{p}_i \in [0,1]^2$), and random orientations,

$$\begin{aligned} x_i &\sim \mathrm{Unif}(0,1),\ i = 1,\dots,n, \\ y_i &\sim \mathrm{Unif}(0,1),\ \ i = 1,\dots,n, \\ \theta_i &\sim \mathrm{Unif}(0,1),\ i = 1,\dots,n, \\ N &= \{(\mathbf{p}_1,\theta_1),(\mathbf{p}_2,\theta_2),\dots,(\mathbf{p}_n,\theta_n)\}. \end{aligned}$$

Assess performance of initial sensor network, $f = C_g(N, \Xi)$.
Set best sensor network and best performance to the initial one, $N_{\mathrm{best}} = N,\ f_{\mathrm{best}} = f$.
for $t = 1,\dots,M$ **do**
  Select a random sensor with uniform distribution, $s \sim \mathrm{DiscUnif}(n)$.
  Apply a random perturbation of sensor position $\mathbf{p}_s = (x_s, y_s)$ and orientation θ_s to generate new candidate sensor network placement N',

$$r_x \sim \mathcal{N}(0,\sigma),\ r_y \sim \mathcal{N}(0,\sigma),\ r_\theta \sim \mathcal{N}(0,\sigma),$$

$$x'_s = x_s + r_x,\ y'_s = y_s + r_y,\ \theta'_s = \theta_s + r_\theta,$$

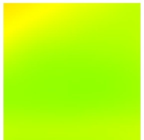
$$N' = \{(\mathbf{p}_1,\theta_1),\dots,(\mathbf{p}'_s,\theta'_s),\dots,(\mathbf{p}_n,\theta_n)\}.$$

  Evaluate performance of new candidate sensor network N', $f' = C_g(N', \Xi)$.
  if $f' > f$, the new sensor network is better than current one, **then**
    Accept new sensor network, $N = N',\ f = f'$.
    if $f' > f_{\mathrm{best}}$, new sensor network is better than best so far, **then**
      Set best sensor network and best performance to the current one, $N_{\mathrm{best}} = N',\ f_{\mathrm{best}} = f'$.
    end if
  else
    Get temperature of current iteration, $E_i = \mathrm{temperature}(t)$.
    if $e_i < E_i,\ e_i \sim \mathrm{Unif}(0,1)$, current sensor network is accepted given the temperature, **then**
      Accept new sensor network, $N = N',\ f = f'$.
    end if
  end if
end for
Return best sensor network found, N_{best}, as final result.

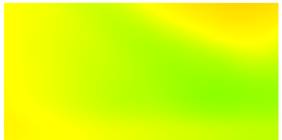
Fig. 5. Pseudo-code of sensor placement with simulated annealing, with perturbation of one sensor position at a time.

D. CMA-ES Evolutionary Algorithm

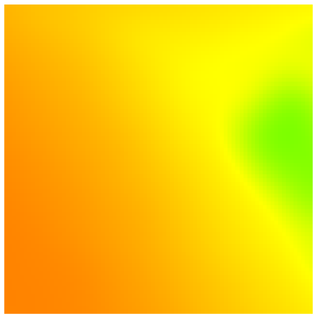
CMA-ES [20] is an evolutionary algorithm known for its good performance and stability [21]. It updates the covariance matrix of the distribution to learn a second order model of the underlying objective function, similar to the approximation of the inverse Hessian matrix in the Quasi-Newton method in classical optimization. However, it does not require partial



(a) Small map



(b) Medium map



(c) Large map

Fig. 6. Two dimensional view of the environment used for placement evaluation by different methods: (a) small map of 45 rows and 45 columns, with elevation ranging from 112.73 m to 115.93 m; (b) medium map of 45 rows and 90 columns, with elevation ranging from 112.63 m to 116.41 m; and (c) large map of 80 rows and 80 columns, with elevation ranging from 112.07 m to 120.30 m.

derivatives.

For sensor placement optimization, the position and orientation of the sensors can be encoded inside an individual, and a population of individuals can be evolved through generations. At the end of the evolution, the individual with the best coverage is chosen as the final solution.

E. Experiments

To conduct our experiments, we selected a mountainous area in North Carolina. The data was provided by a raster layer map in the "OSGeo Edu" dataset[1], that stores geo-spatial information about parts of the North Carolina State, USA. More specifically, we focus on a portion of the map that covers a small watershed in a rural area near NC capital city, Raleigh. The coordinate system of the map is the NC State Plane (Lambert Conformal Conic projection), metric units and NAD83 geodetic datum. We used three portions of the map for our experiments (See Figure III-E). A small map with 2025 cells to be covered by 3 sensors, a medium map with 4050 cells to be covered by 6 sensors, and a large map with 6400 cells to be covered by 9 sensors.

Sensors are modelled following a description given in Sec. II, using as parameters $\alpha = 350$, $\beta = 10$, and $\omega = 3$. For all methods except for the deterministic approach, each sensor placement optimization scheme was run 10 times, from which are estimated the average and the standard deviation of each method. CPU times are also averaged over the 10 runs, in order to compare the resources required by each method to produce a solution. These time values have been evaluated by running the methods on one core of Intel I7 computers clocked at 2.8 GHz.

[1]Available at http://grass.itc.it/download/data.php.

TABLE I

COVERAGE PERCENTAGE ON THE TARGET AREAS WITH VARIOUS NUMBER OF SENSORS. EACH SCHEME HAS BEEN RUN 10 TIMES, WITH COVERAGE LOSS AVERAGES AND THE CORRESPONDING STANDARD DEVIATIONS REPORTED. NOTE THAT 100% COVERAGE IS NOT POSSIBLE WITH A FINITE NUMBER OF SENSORS, GIVEN ITS PROBABILISTIC NATURE.

Method	Small map	Med. map	Large map
Number of sensors	3	6	9
Search dimensions	9	18	27
Deterministic	74.42%	81.68%	71.90%
SA average	86.07%	84.75%	80.48%
SA stdev	0.71%	4.18%	3.26%
SA CPU time	7 min.	12 min.	22 min.
L-BFGS average	84.69%	83.64%	78.98%
L-BFGS stdev	1.86%	1.99%	2.67%
L-BFGS CPU time	15 min.	87 min.	186 min.
CMA-ES average	87.76%	90.97%	89.65%
CMA-ES stdev	0.65%	0.75%	0.68%
CMA-ES CPU time	60 min.	133 min.	285 min.

For simulated annealing the perturbations for positions and for orientation is a Gaussian distribution with standard deviation σ_{sa}. The optimal value for σ_{sa} is found by trial and error, so that the optimal value for each map was found independently. These optimal values are $\sigma_{sa} = 0.02$ for the small map, $\sigma_{sa} = 0.02$ for the medium map, and $\sigma_{sa} = 0.08$ for the large map. For each run, SA last 350 iterations, which is more than necessary to reach convergence of the algorithm to a solution. With L-BFGS, a history of the $m = 10$ past updates of the position and gradient are used to limit the memory usage, and the stop criteria is parametrized by values $factr = 10^7$ and $pgtol = 1.0^{-5}$. CMA-ES is run with the a population of $\mu = 6$ parents and $\lambda = 13$ offspring, for 350 generations, which was enough to achieve convergence. A mutation factor $\sigma = 0.167$ is also used.

All optimization programs are written in Python. We used the implementation of L-BFGS from the well-known SciPy library[2]. CMA-ES implementation was taken from DEAP[3], a Python library for evolutionary algorithms developed at Université Laval.

F. Experimental Results

We compare the performance of the four mentioned placement methods, that is Deterministic approach, Simulated Annealing, L-BFGS, and CMA-ES.

We ran each optimization scheme 10 times and calculated the corresponding coverage loss percentage on the target areas. A qualified sensor optimization scheme should have low coverage loss and low standard deviations of coverage loss given a number of runs. In other words, we are evaluating each algorithm in terms of both accuracy and robustness. The results of each method using the best parameter sets are shown in the Table I.

Among all tested methods, CMA-ES has by far the best performance, in terms of both low coverage loss and low

[2]Available at http://www.scipy.org.

[3]Available at http://deap.googlecode.com.

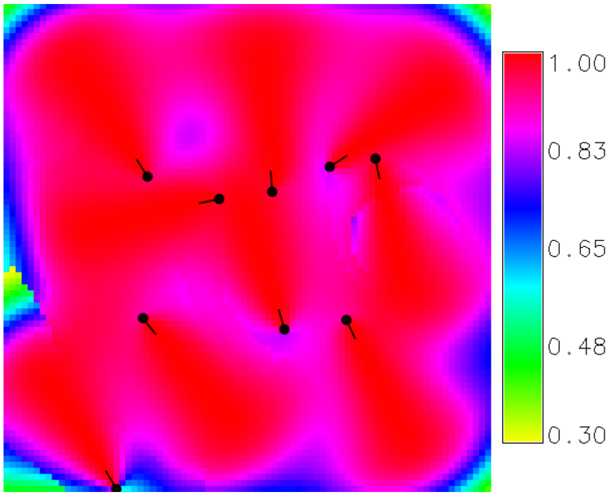


Fig. 7. Example of a sensor placement optimization result on the large map using CMA-ES.

standard deviation of coverage loss. This is not surprising, because CMA-ES performs a parallel search in the search space and unlike simulated annealing combines the results of several stochastic search paths to find the optimal solution. It is however the most expensive optimization in terms of computation resources required, as the number of tested solutions is much higher than with the other methods (4550 solutions evaluated by runs, that is 13 offspring over 350 generations). Figure 7 presents a sample result of sensor placement using CMA-ES on a large map.

IV. Conclusion

A framework for optimization of sensor placement is proposed in this paper. The novelty of this framework lies in the integration of terrain information (elevation maps) with a probabilistic sensor model. Different optimization methods have been tested with this framework, with the results reported showing a variability in the performances. The CMA-ES optimization method clearly outperformed the two others (SA and L-BFGS) that showed similar levels of performance. This demonstrates that the optimization problem as defined in the current framework is quite a difficult one, requiring a sophisticated parallel stochastic search method.

From a modelling perspective, we agree that our proposal has room for refinement, for example by simulating signal propagation. However, our point is to make a proof of concept of sensor placement through the use of black-box optimization, using sophisticated probabilistic modelling of sensors operating in a given environment. If one has better models, our proposed optimization approach should still be applicable.

This serves as a starting point to further investigate the use of evolutionary algorithms in sensor placement optimization. Beside these, another potential future work will be the multi-objective optimization for sensor deployment, given multiple concerns such as number of sensors used, energy saving, and multiple coverage.

Acknowledgements

This project is funded by the GEOIDE Network of Centres of Excellence (Canada) and MDA Systems Ltd. We also thank Albert Hung-Ren Ko for his early participation in the project and Annette Schwerdtfeger for proofreading the manuscript.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] Y. C. Wang and Y. C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed System*, vol. 19, no. 9, pp. 1280 – 1294, 2008.

[3] S. B. K. Kar, "Node placement for connected coverage in sensor networks," *Proceedings of the Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[4] H. G. Z. Zhou, S. Das, "Connected k-coverage problem in sensor network," *Proceedings of 13th International Conference on Computer*, vol. 1, 2007.

[5] J. B. S. Kumar, T. H. Lai, "On k-coverage in a mostly sleeping sensor network," *Wireless Network*, vol. 14, pp. 277 – 294, 2006.

[6] B. Liu and D. Towsley, "A study of the coverage of large-scale sensor networks," in *Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS04)*, 2004, pp. 475–483.

[7] M. Hefeeda and H. Ahmadi, "Energy efficient protocol for deterministic and probabilistic coverage in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, pp. 579–593, 2009.

[8] J. V. Nickerson and S. Olariu, "Protecting with sensor networks: Attention and response," *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, 2007.

[9] S. Olariu and J. V. Nickerson, "Protecting with sensor networks: Perimeters and axes," *MILCOM*, 2005.

[10] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 131–142, 2006.

[11] W. B. H. M. M. Holland, R. G. Aures, "Experimental investigation of radio performance in wireless sensor network," *Proceedings of IEEE SECON*, 2006.

[12] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Transaction on Computers*, vol. 54, no. 8, pp. 978–991, 2005.

[13] N. Ahmed, S. S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, pp. 672–681, 2005.

[14] A. Boukerche and X. Fei, "A coverage-preserving scheme for wireless sensor network with irregular sensing range," *Ad Hoc Network*, vol. 5, no. 8, pp. 1303–1316, 2007.

[15] H. Topcuoglu, M. Ermis, and M. Sifyan, "Positioning and utilizing sensors on a 3-d terrain part i—theory and modeling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, no. 99, pp. 1–7, 2010.

[16] V. Akbarzadeh, A. Ko, C. Gagné, and M. Parizeau, "Topography-Aware Sensor Deployment Optimization with CMA-ES," *Parallel Problem Solving from Nature–PPSN XI*, pp. 141–150, 2010.

[17] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2006, pp. 131–142.

[18] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, p. 671, 1983.

[19] J. Nocedal and S. Wright, *Numerical optimization*. Springer verlag, 1999.

[20] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159 – 195, 2001.

[21] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 6, no. 15, pp. 617–644, 2009.